Analyzing Early-Stage Venture Capital Investment to pick "Winning" Companies

By Cole Crescas, James Cook, Aldo Pioline Antony Charles

Introduction:

Venture Capital is a domain of Investment Banking specialized in backing early stage (pre-IPO) companies. Unlike the traditional stock market (NASDAQ, S&P, etc.), retail investors are barred from investing in them and is reserved exclusively for these kinds of banks or highly wealthy individuals. We hear of 'Unicorns' all the time that ballooned into huge billion-dollar companies, but startups carry substantial risk and only the very best are able to become the next Amazon, Tesla, Apple, or Microsoft. Another challenge with startups is that they are very illiquid, unable to easily get out of your position as there are very few 'exit' opportunities and can come at high costs. Correctly classifying such high risk, high reward companies is an ongoing and fundamental problem to investment banking. Previous research has been conducted from Imperial College, using Startup data from various countries which we have expanded upon in this report.

Problem Statement:

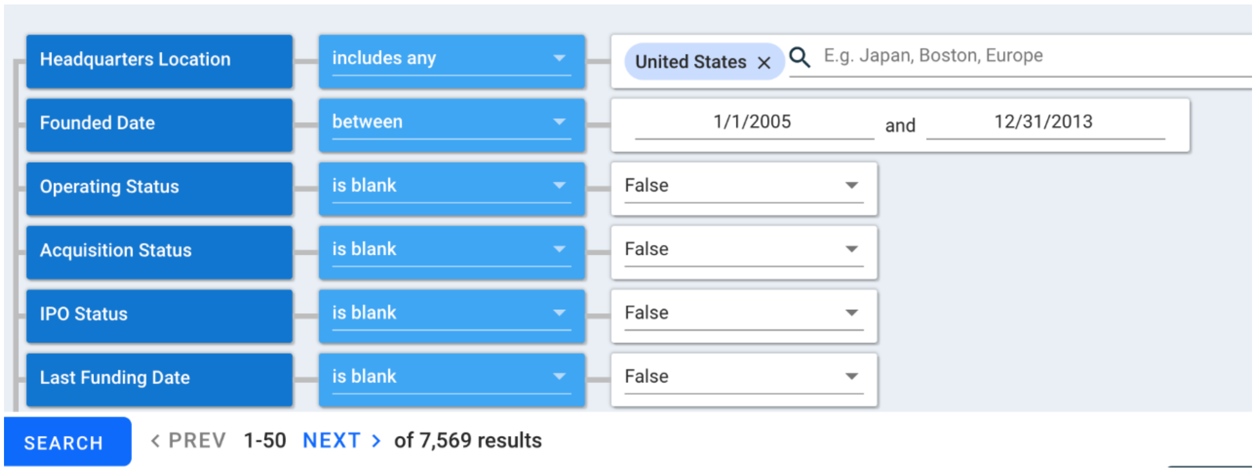
The goal of the project is to analyze around 7,500 US based companies founded between 2005 & 2013 by building a model to correctly classify successful & unsuccessful companies based on our set criteria of success. After reading past research projects tackling a similar problem, we decided to base our success criteria on assuming a company was continuing to grow or has gone public [1][2]. We defined this by setting our search parameters to having any of the following conditions:

- Has had a funding round in 2018 or later.
  - Assumes that the company is still being funded and growing.
- Has had an IPO.
  - Once public, an Exit Opportunity is always available and assumes success.
- Has had more than 7 funding rounds.
  - Indicates that the startup still has potential and growth.

Methods

Data Collection:

The collected data for this project was sourced from Crunchbase [3]. Crunchbase is one of the leading databases for private companies with info from various industries around the world. We choose to analyze US companies that were founded between 2005 & 2013 and made sure the features operating status, acquisition status, IPO Status and last funding date had no missing values. This was done to ensure that our success criteria could be built around existing data. Our resulting dataset was 7,569 companies with 123 features ranging from Company Description to Estimated Revenue Range.



Headquarters Location	includes any	United States ✕ E.g. Japan, Boston, Europe
Founded Date	between	1/1/2005 and 12/31/2013
Operating Status	is blank	False
Acquisition Status	is blank	False
IPO Status	is blank	False
Last Funding Date	is blank	False

Figure 1: Crunchbase Query Parameters & results

Data Pre-Processing:

To get our dataset into a cleaner format, we removed features that were missing more than half of the data. We then manually removed features that were not as important such as twitter URL, number of apps or USD conversion columns. Different features were then converted to strings or floats for analysis purposes. In order to get our data frame to be continuous data, the features Acquisition Status, IPO status & Estimated Revenue Range were converted using one hot encoding. The class variable was also added based on our predetermined success criteria. We also wanted to include industry groups into our model, so the 'industry group' feature was converted to a tf-idf bag of words format. We also tried experimenting topic modelling using Latent Dirichlet Allocation (LDA) to group various industry groups into common topics but nevertheless it did not improve model's success and moreover performing LDA was time consuming and tf-idf was faster, so we stuck with it. Finally, we used under sampling from the imblearn package to ensure our model had a 50/50 split between classes. This was used to improve our test set recall score. We also held out the last 15 companies, containing both successful and unsuccessful companies, to validate our model's effectiveness later.

Exploratory Data Analysis:

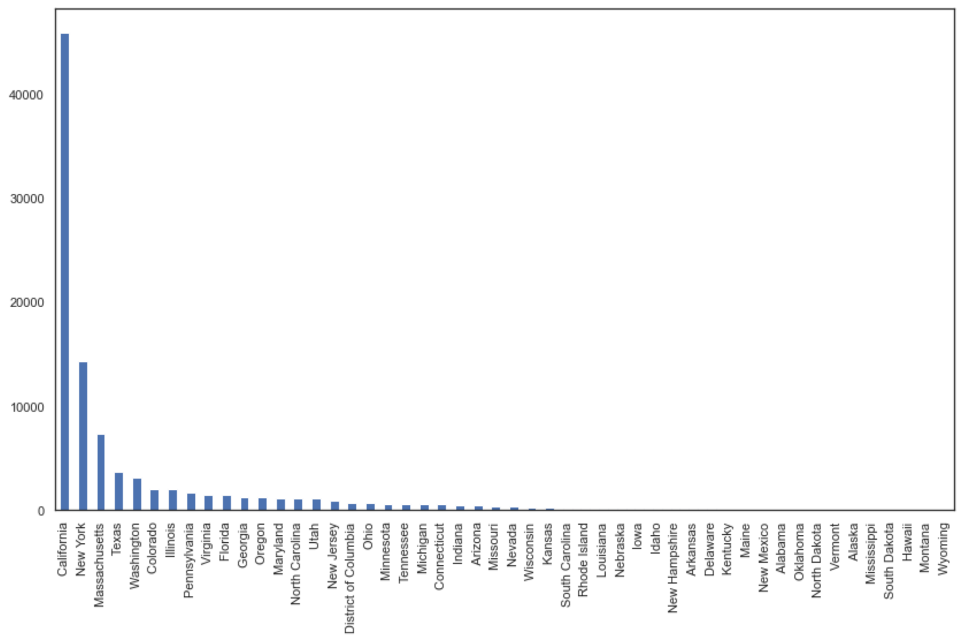


Figure 2: Frequency distribution of startups over different locations (USA)

Creating a simple bar chart, it is clearly seen that California, New York, and Massachusetts are the main hubs of startups, with California having by the far the most. This holds true to our expectations, as California is known for Silicon Valley, churning out startups left and right.

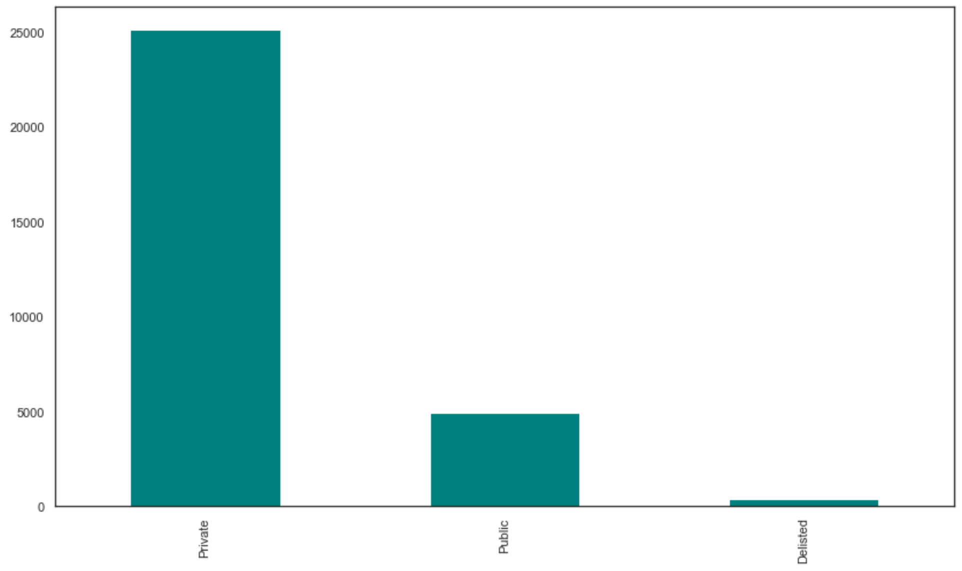


Figure 3: Frequency of Startups' IPO status

Next, we explored the distribution of startups' IPO status as it is a vital criterion for defining success in our case and as we can see, majority of the companies are still private while some of them are public which goes well along with real-world statistics also proving the validity of our data. A very small number of startups got delisted and are interesting data points for further investigation.

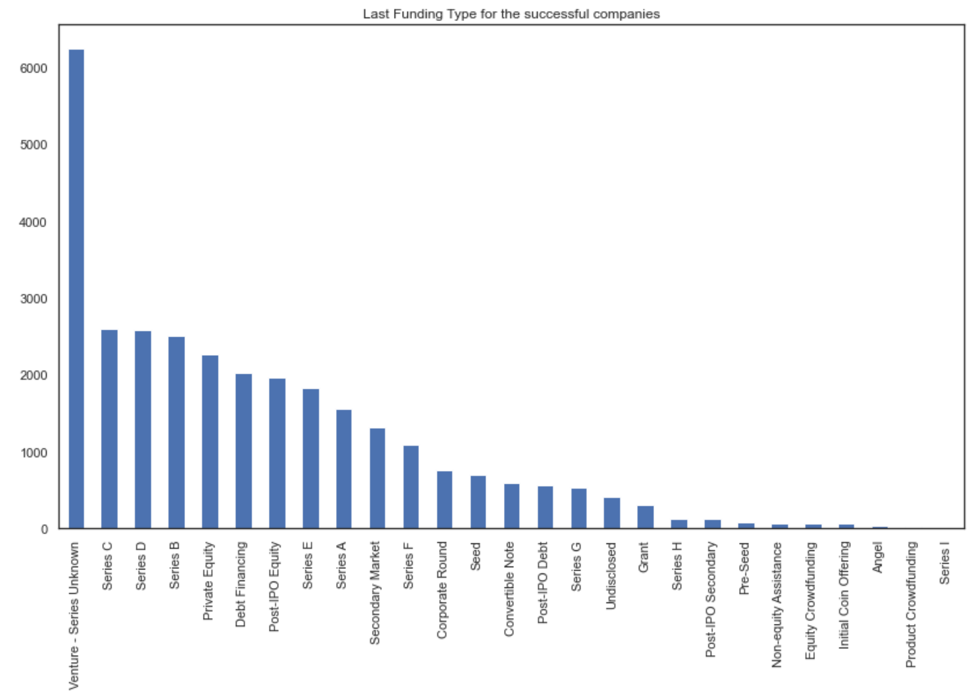


Figure 4: Distribution of Startups' last funding round

We further explored the maximum round of funding each startup has received. Since this is a manually entered data, most of the series funding is unknown. Digging further we can see that most of the startups have ended up with a Series-C, Series-D and Series-B and companies that go beyond these points decline in numbers which is seen in the histogram.

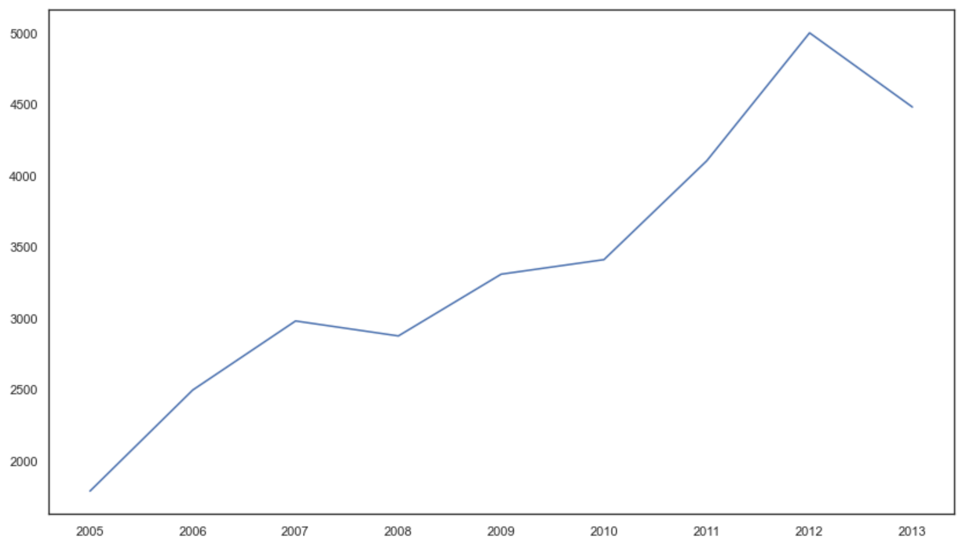


Figure 5: Investment trend over 8 years

The above line graph is very important in showing the trend of investments overtime between the years 2005 to 2013. As observed, we can see a drop in the trend during the years 2007 and 2012 which were two important years when the financial crisis happened due to different reasons and startups struggled at that time to get investments. This trend can be further explored and used for time series model to determine how successful a startup can be.

After defining our success criteria, we were interested in exploring the top investment firms based on the number of successful investments and these were the top 5 investors:

[' SV Angel', ' Andreessen Horowitz', 'Goldman Sachs', ' New Enterprise Associates', 'TechStars']

Using Pandas' grouping technique we were able to get the top sectors which had successful investments as follows:

[Software, SaaS, Enterprise Software, Information Technology, Health Care, Mobile, Analytics]

Next, we proceeded to determine the frequency distribution of companies that ended up getting acquired vs the ones that made acquisitions and we were able to observe the below.

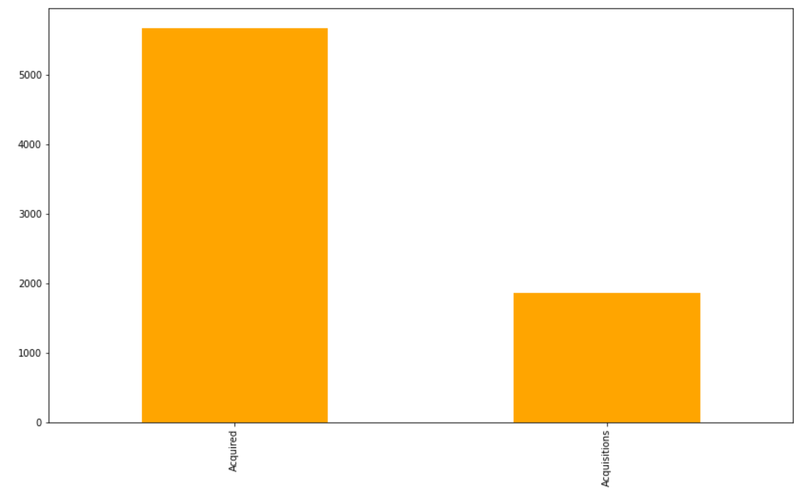


Figure 6: Frequency of Startups' Acquisition status

As we can see, majority of the companies ended up getting acquired while some of them go public and make acquisitions. The metric above is also based on the success criteria that we defined which seem to hold the dataset along the correct trend as the real-world data.

Models:

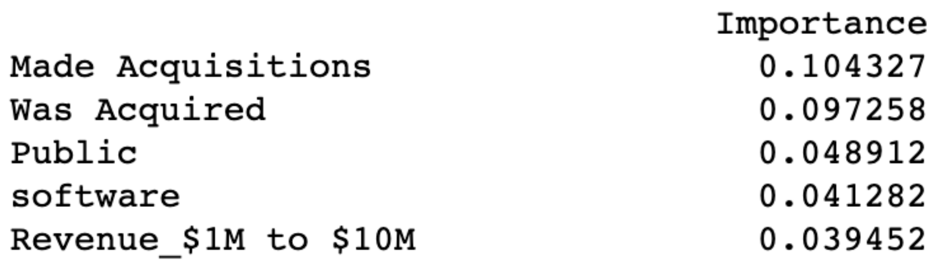
Startup success prediction:

We tried various models on our dataset with the aim of predicting successful companies while not overfitting the data. Model complexity and training time were also factors in this case as we wanted to be able to iterate on our hyperparameter selection. Following model creation for each algorithm we analyzed the model accuracy on both the training & test set as well as graphed the confusion matrix to determine which models would be the best selection for our use case. Outlined below is the description of each of the algorithms we used as well as some of the hyperparameters for each case.

In addition, we used a grid search approach to determine the best test & training split optimizing by the combined test & training set recall score.

Random Forest:

We implemented the random forest algorithm from sklearn to implement the base decision tree learning on our dataset. In addition, we were looking to utilize the feature importance attribute from random forest. This was able to tell us the most important features used to split the data. Using this feature list, the top 5 most important features were:



```
                                   Importance
Made Acquisitions                    0.104327
Was Acquired                         0.097258
Public                               0.048912
software                             0.041282
Revenue_$1M to $10M                  0.039452
```

Figure 7: Feature Importance using Random Forrest

We used bootstrapping as well as the Out of Bag Score to discover which features had the most importance. Our Out of Bag Score was 0.7482, which was close to our training & test accuracy.

Using the most important features, we were able to create a list of private companies that fulfilled the important success criteria. The aim of this was to create a list of companies ready for late-stage venture capital investment. Using the industry criteria & revenue range a sample of our list is attached here:

'UiPath', 'Automattic', 'Interos', 'appfire', 'Bossa Nova Robotics', 'DialogTech', 'Nayax', 'Performive', 'RockYou', 'Persivia'

This would be used by a VC firm to select the next winners based on past success.

Neural Net:

Following our Random Forest model, we aimed to implement the MLPClassifier from sklearn. We chose 100 hidden layers, and 50 maximum iterations, with early stopping set to true. After 11 iterations our model stopped improving and was stopped with a validation score of .728814. We chose to use the default SGD loss function. We knew that neural networks generally outperform less complex models on a large data set. Our dataset could be classified as medium sized in this case, but our best results were from the Neural Net model.

Logistic Regression:

We used our logistic regression model as a baseline comparison for other models' performance. Since this is a simpler model than others but works very well on our continuous dataset.

Boosting

We wanted to implement boosting techniques on our dataset to determine if this method would improve results by compiling weak classifiers. We started with the AdaBoost algorithm to try and improve our model's performance with multiple weak classifiers. For this case we chose to use the default parameters with 100 estimators. Getting similar results to other models we then tried to use gradient boosting. For our gradient boosting algorithm, we chose to use 100 estimators, a learning rate of 1, and a max depth of 1. Both boosting algorithms provided similar results with gradient boosting slightly outperforming AdaBoost in this case. A full table of results can be seen in the results section for each algorithm.

K-Nearest Neighbors:

KNN is a lazy learner & nonparametric model which is heavily affected by the hyperparameters k & c. We chose to use the 7 nearest neighbors & the default c value. KNN performed better than logistic regression on the training set but not on the test set likely due to less data to learn from.

Ensemble – Soft Voting & Hard Voting:

We combined 7 estimators in this ensemble. The 7 estimators in this case were: Random Forest, Nueral Net, LinearSVC, Logistic Regression, AdaBoost, Gradient Boosting & KNN. We choose to use the default parameters for both Soft & Hard voting. Soft voting overall performed better than hard voting due to the added flexibility.

Acquisition model:

Acquisition status of a company determines the success of an investor's portfolio and their returns. Predicting if a startup will get acquired or not helps the investor(s) to either diversify their investments or plan their entry vs exit strategies. So, once we defined a success criterion and tested it, we were interested in seeing if we can use the success criteria along with other original and engineered features to predict the acquisition status of a company. We used the correlation matrix to determine the appropriate features that contribute towards the column, '*Acquisition Status*' and engineered a few of them to suite our needs which included modifying the founding date of the company into the age in years and a similar calculation with respect to the last funding date. After performing the same preprocessing steps as the previous model, we tried out different algorithms to see if it was possible to predict the acquisition status of a company [4].

The modelling algorithms used here are like the previous model with similar hyperparameters with the addition of few other techniques detailed later in this section. The following observations were made.

Logistic Regression:

Logistic regression did not perform well on the dataset, and we can see the low accuracy score for both the training and test dataset. This is mostly because the features are not linearly dependent or separable which was observable in the correlation matrix. So, the model was not suitable for predicting the acquisition status of the company. The result can be found in the Results section.

K-Nearest Neighbors:

The KNN model performed slightly better than the logistic regression model with higher accuracy, but comparatively lower recall score and the model seems to understand the dataset. We used 7 neighbors like the success model parameter here and with different K values, the model might perform differently. But still the model was not reliable because of the lower score.

Random Forest:

The success model previously discussed seemed to perform good on tree-based modelling algorithms. So, we attempted different flavors of decision tree techniques here. First, we tried the **Random Forest** algorithm. It produced a 100% training accuracy and training recall score which raised a doubt if it was overfitting, but when it was run over new data, it was able to predict the right acquisition status with good accuracy and was able to construct proper decision boundaries and was not overfitting.

AdaBoost:

AdaBoost, a learning model that adapts based on the misclassified datapoints performed well on the dataset with the scores shown in the results also because of the nature of the data making them easily separable by decision trees.

Gradient Boosting:

Proceeding further, we wanted to experiment how introducing a loss function and improving the model by minimizing the loss would improve the recall and accuracy overall. We tried Gradient Boosting with *n_estimators=100, learning_rate=1, max_depth=1, random_state=0* and it gave slightly improved test recall score than AdaBoost but nothing significant. But the overall accuracy and recall seemed good ~90%

XGBoost:

XGBoost, a popular gradient boosting method adopted for high dimension data was used here and its performance was lower than AdaBoost as well. While XGBoost is a powerful algorithm by itself in terms of adapting to the previous errors, it quite did not suite the dataset as XGBoost is suitable for higher dimensional data and since we have a limited feature space, XGBoost did not give the best performance that it was developed for, but the results were good.

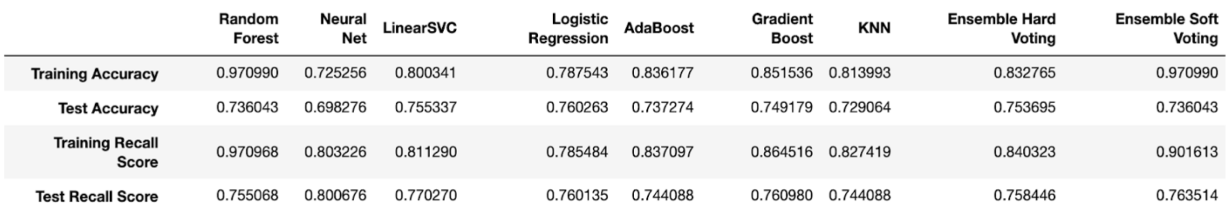
Ensemble Learning:

Ensemble hard and soft voting were performed using random forest, gradient boosting, AdaBoost and KNN estimators and they performed very well with good accuracy and recall scores. The Soft voting classifier's training and test accuracy scores were like the results from the random forest model which could be due to the high probability decisions made by random forest in this case.

Results:

Success Prediction Model:

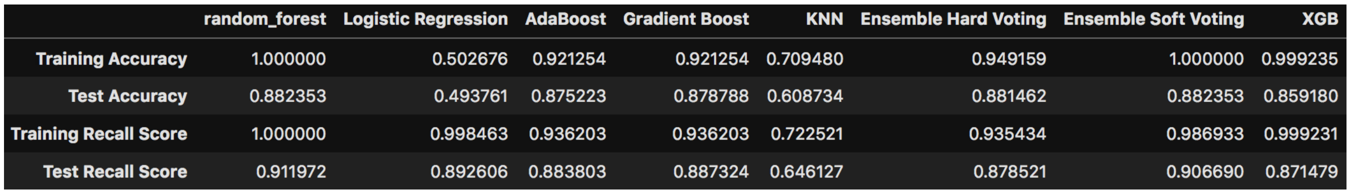
Our primary metric for evaluating model success is the recall score or true positive rate. Venture Capital firms expect to take some losses as long as they can correctly select successful companies that are able to far exceed any losses. Therefore, recall matters much more than Precision so that an opportunity, especially a unicorn like Facebook, Uber or UiPath isn't missed.



	Random Forest	Neural Net	LinearSVC	Logistic Regression	AdaBoost	Gradient Boost	KNN	Ensemble Hard Voting	Ensemble Soft Voting
Training Accuracy	0.970990	0.725256	0.800341	0.787543	0.836177	0.851536	0.813993	0.832765	0.970990
Test Accuracy	0.736043	0.698276	0.755337	0.760263	0.737274	0.749179	0.729064	0.753695	0.736043
Training Recall Score	0.970968	0.803226	0.811290	0.785484	0.837097	0.864516	0.827419	0.840323	0.901613
Test Recall Score	0.755068	0.800676	0.770270	0.760135	0.744088	0.760980	0.744088	0.758446	0.763514

From the above table, the Neural Net performs the best on the Test set with very similar scores to the Training. With such a small difference between the two scores, it indicates that this model is not overfitting to the training set and is a good model to use. Neural nets often outperform other models, especially as the number of features & data points increase.

Acquisition Model:



	random_forest	Logistic Regression	AdaBoost	Gradient Boost	KNN	Ensemble Hard Voting	Ensemble Soft Voting	XGB
Training Accuracy	1.000000	0.502676	0.921254	0.921254	0.709480	0.949159	1.000000	0.999235
Test Accuracy	0.882353	0.493761	0.875223	0.878788	0.608734	0.881462	0.882353	0.859180
Training Recall Score	1.000000	0.998463	0.936203	0.936203	0.722521	0.935434	0.986933	0.999231
Test Recall Score	0.911972	0.892606	0.883803	0.887324	0.646127	0.878521	0.906690	0.871479

Comparing the different models used to predict if a company would end up getting acquired or not, Random Forest performed better than other techniques especially better than the Boosting techniques which is an interesting observation because of the nature of the dataset, combining different aggregation of features that would work well rather than trying different methods to boost the prediction which is what the boosting techniques such as Gradient Boosting, AdaBoost and XGBoost did. The results of Random Forest were like Soft Voting classifier in terms of accuracy, and it makes sense because both the methods make decision by aggregation, but Random Forest has a higher recall score of 91.1% which is important for us in the current use case. So ideally, Random Forest performed better overall and suited our use case.

Discussion:

Our model can successfully classify companies based on preset metrics with a high degree of success in accuracy compared to the baseline of other research papers (~7% better). One reason for this difference is that most papers do not have the same focus on US companies like we do, and instead use international data. While there can also be successful startups abroad, comparing companies that don't face the same regulatory risk, entrepreneurial support, or other outside factors, is not an apples-to-apples comparison and the model will perform worse overall. Most Venture Capitalists will have an area of expertise, and this model could be used to further aid them in surveying US based startups.

Also, the acquisition model was able to predict with good recall and accuracy about a company's acquisition status using the funding details, funding rounds and other factors. Although this is good enough, predicting acquisition requires more processes like time series prediction and other algorithmic and analytical techniques which will be explored in the future.

Resources:

[1]https://www.imperial.ac.uk/media/imperial-college/faculty-of-natural-sciences/department-of-mathematics/math-finance/HENGSTBERGER_THOMAS_01822754.pdf

[2]https://scholar.harvard.edu/files/saghafian/files/00_startup_ml.pdf

[3]https://www.crunchbase.com/

[4] https://www.cs.cmu.edu/~guangx/papers/icwsm12-short.pdf

Appendix:

GitHub link: https://github.com/colaso96/Predicting_Venture_Capital_Investment_Success